

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Yoshiki Okumura

Application No.: NEW

Group Art Unit: Not Yet Assigned

Filed: February 20, 2004

Examiner: Not Yet Assigned

For: ARITHMETIC DEVICE FOR MULTIPLE PRECISION ARITHMETIC FOR
MONTGOMERY MULTIPLICATION RESIDUE ARITHMETIC

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith
a certified copy of the following foreign application:

Japanese Patent Application No(s). 2003-46527

Filed: February 24, 2003

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date(s) as evidenced by the certified papers attached hereto, in accordance with the
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: February 20, 2004

By: 

J. Randall Beckers
Registration No. 30,358

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

JAPAN PATENT OFFICE

This is to certify that the annexed is a true copy of the following application as filed with this office.

Date of Application: February 24, 2003

Application Number: Patent Application No. 2003-046527
[ST.10/C] [JP2003-046527]

Applicant(s): FUJITSU LIMITED

October 14, 2003

Commissioner,

Japan Patent Office Yasuo IMAI

Certificate No.P2003-3084342

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 2 月 2 4 日
Date of Application:

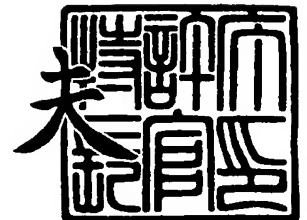
出 願 番 号 特 願 2 0 0 3 - 0 4 6 5 2 7
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 0 4 6 5 2 7]

出 願 人 富 士 通 株 式 会 社
Applicant(s):

2 0 0 3 年 1 0 月 1 4 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康



出証番号 出証特 2 0 0 3 - 3 0 8 4 3 4 2

【書類名】 特許願

【整理番号】 0253273

【提出日】 平成15年 2月24日

【あて先】 特許庁長官殿

【国際特許分類】 G09C 1/00
G06F 7/72
H04L 9/00

【発明の名称】 モンゴメリ乗算剰余の多倍長演算のための演算装置

【請求項の数】 5

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 株式会社富士通コンピュータテクノロジー内

【氏名】 奥村 嘉樹

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100074099

【住所又は居所】 東京都千代田区二番町8番地20 二番町ビル3F

【弁理士】

【氏名又は名称】 大菅 義之

【電話番号】 03-3238-0031

【選任した代理人】

【識別番号】 100067987

【住所又は居所】 神奈川県横浜市鶴見区北寺尾7-25-28-503

【弁理士】

【氏名又は名称】 久木元 彰

【電話番号】 045-573-3683

**【手数料の表示】****【予納台帳番号】** 012542**【納付金額】** 21,000円**【提出物件の目録】****【物件名】** 明細書 1**【物件名】** 図面 1**【物件名】** 要約書 1**【包括委任状番号】** 9705047**【プルーフの要否】** 要

【書類名】 明細書

【発明の名称】 モンゴメリ乗算剰余の多倍長演算のための演算装置

【特許請求の範囲】

【請求項 1】 ビットパターンで表された被乗数 A と乗数 B の乗算を行う演算装置であって、

被乗数 A から 2 次 Booth アルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数 B を前記 2 次 Booth アルゴリズムによりエンコードして、乗数 B の連続する 3 つのビットである b_{2i+1} 、 b_{2i} 、および b_{2i-1} を指定する i の値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、 i の数と同じ数だけの部分積を加算して、乗算結果を生成する加算手段とを備え、

前記エンコーダ手段が、 i が 0 のときには $-A$ を表す部分積を選択するための選択信号を出力し、 i が 0 以外のときには 0 を表す部分積を選択するための選択信号を出力し、前記加算手段が、該 $-A$ を表す部分積から被乗数 A の 2 の補数を生成して、該被乗数 A の 2 の補数を前記乗算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【請求項 2】 ビットパターンで表された被乗数 A と乗数 B の乗算を行う演算装置であって、

被乗数 A から 2 次 Booth アルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数 B を前記 2 次 Booth アルゴリズムによりエンコードして、乗数 B の連続する 3 つのビットである b_{2i+1} 、 b_{2i} 、および b_{2i-1} を指定する i の値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、 i の数と同じ数だけの部分積を加算して、乗算

結果を生成する加算手段とを備え、

前記エンコーダ手段が、 i が 0 のときには $-A$ を表す部分積を選択するための選択信号を出力し、 i が 0 以外のときには 0 を表す部分積を選択するための選択信号を出力し、前記加算手段が、該 $-A$ を表す部分積から被乗数 A の 1 の補数を生成して、該被乗数 A の 1 の補数を前記乗算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【請求項 3】 ビットパターンで表された被乗数 A と乗数 B を乗算し、乗算結果とビットパターンで表された数 C および数 D とを加算する、乗算加算を行う演算装置であって、

被乗数 A から 2 次 Booth アルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数 B を前記 2 次 Booth アルゴリズムによりエンコードして、乗数 B の連続する 3 つのビットである b_{2i+1} 、 b_{2i} 、および b_{2i-1} を指定する i の値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、 i の数と同じ数だけの部分積と数 C と数 D とを加算して、乗算加算結果を生成する加算手段とを備え、

前記エンコーダ手段が、 i が 0 のときには $-A$ を表す部分積を選択するための選択信号を出力し、 i が 0 以外のときには 0 を表す部分積を選択するための選択信号を出力し、前記加算手段が、該 $-A$ を表す部分積から被乗数 A の 1 の補数を生成し、該被乗数 A の 1 の補数と数 C と数 D とを加算した結果を前記乗算加算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【請求項 4】 k ビットのビットパターンで表された被乗数 A と乗数 B を乗算し、乗算結果と k ビットのビットパターンで表された数 C および数 D とを加算することで、1 ブロックが k ビットである $g+1$ 個のブロックからなる整数 Y から 1 ブロックが k ビットである g 個のブロックからなる整数 N を減算する演算を行う演算装置であって、

被乗数 A から 2 次 Booth アルゴリズムにおける複数の部分積を生成する部

分積生成手段と、

乗数Bを前記2次Boothアルゴリズムによりエンコードして、乗数Bの連続する3つのビットである b_{2i+1} 、 b_{2i} 、および b_{2i-1} を指定する i の値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、 i の数と同じ数だけの部分積と数Cと数Dとを加算して、 $2k$ ビットの乗算加算結果を生成する加算手段と、

前記乗算加算結果の一部のビットを反転する反転手段とを備え、

前記部分積生成手段が、整数Nの j 番目のブロック n_j を被乗数Aとして用い、前記エンコーダ手段が、 i が0のときには $-A$ を表す部分積を選択するための選択信号を出力し、 i が0以外のときには0を表す部分積を選択するための選択信号を出力し、前記加算手段が、該 $-A$ を表す部分積から被乗数Aの1の補数を生成し、 $j-1$ 番目のブロックの乗算加算からのキャリーを数Cとして用い、整数Yの j 番目のブロック y_j を数Dとして用いて、該被乗数Aの1の補数と数Cと数Dとを加算した結果を j 番目のブロックの乗算加算結果として出力し、前記反転手段が、該 j 番目のブロックの乗算加算結果の一部のビットを反転して $j+1$ 番目のブロックの乗算加算へのキャリーを生成することを特徴とする演算装置。

【請求項5】 ビットパターンで表された整数IおよびJと剰余の法Nを1ブロックが k ビットである g 個のブロックにそれぞれ分割して、 $Y = I J 2^{-kg} \bmod N$ となるモンゴメリ乗算剰余の多倍長演算を行う演算装置であって、

k ビットの被乗数A、乗数B、数C、および数Dのそれぞれについて、与えられた複数の値のいずれかを選択して出力する第1の選択手段と、

前記第1の選択手段から出力される被乗数Aから2次Boothアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

前記第1の選択手段から出力される乗数Bを前記2次Boothアルゴリズムによりエンコードして、乗数Bの連続する3つのビットである b_{2i+1} 、 b_{2i}

i 、および b_{2i-1} を指定する i の値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する第2の選択手段と、

前記第2の選択手段から出力される、 i の数と同じ数だけの部分積と、前記第1の選択手段から出力される数 C と数 D とを加算して、 $2k$ ビットの乗算加算結果を生成する加算手段と、

前記乗算加算結果の一部のビットを反転する反転手段とを備え、

前記第1の選択手段が、整数 N の j 番目のブロック n_j を被乗数 A として選択し、 $j-1$ 番目のブロックの乗算加算からのキャリーを数 C として選択し、 Y の j 番目のブロック y_j を数 D として選択し、前記エンコーダ手段が、 i が0のときには $-A$ を表す部分積を選択するための選択信号を出力し、 i が0以外のときには0を表す部分積を選択するための選択信号を出力し、前記加算手段が、該 $-A$ を表す部分積から被乗数 A の1の補数を生成し、該被乗数 A の1の補数と数 C と数 D とを加算した結果を j 番目のブロックの乗算加算結果として出力し、前記反転手段が、該 j 番目のブロックの乗算加算結果の一部のビットを反転して $j+1$ 番目のブロックの乗算加算へのキャリーを生成するような動作モードを設けたことを特徴とする演算装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、モンゴメリ乗算剰余の多倍長演算を行うための演算装置に関する。

【0002】

【従来の技術】

近年、コンピュータネットワーク上では、セキュリティ対策として、暗号化データ通信やユーザ認証（デジタル署名）が多く使用されている。これらのセキュリティ対策では、RSA（Rivest-Shamir-Adleman）暗号や楕円曲線暗号等の暗号処理が利用される。

【0003】

R S A 暗号や楕円曲線暗号は非常に大きいビット数の数値を扱うため、これらの暗号処理で使われるモンゴメリ乗算剰余の演算では、ソフトウェアやハードウェアで実現し易い多倍長演算アルゴリズムがよく使われる。多倍長演算アルゴリズムとは、非常に大きいビット数の数値を、ソフトウェアやハードウェアで扱い易いビット数（ブロック）に分割して、ブロック単位の演算を繰り返すアルゴリズムである。

【0004】

モンゴメリ乗算剰余の多倍長演算アルゴリズムとしては、図12のようなアルゴリズムが知られている（例えば、特許文献1参照）。図12において、AおよびBは整数であり、Nは剰余の法であり、Yは演算結果である。図12の左側に示すアルゴリズムは、その右側に示すように変形することができる。このアルゴリズムは、 $Y = A B 2^{-k} g \bmod N$ の演算を表しており、①～⑥の処理からなる。

【0005】

g はA、B、およびNを k ビット＝1ブロックで分割した時のA、B、およびNのブロック数であり、Yのブロック数は $g+1$ である。 a_i 、 b_i 、 n_i 、および y_i はそれぞれA、B、N、およびYの*i*番目のブロックを表す。A、B、N、およびYは、それぞれ、 a_i 、 b_i 、 n_i 、および y_i を用いて次のように表される。

【0006】

$$A = (a_{g-1}, a_{g-2}, \dots, a_1, a_0)$$

$$B = (b_{g-1}, b_{g-2}, \dots, b_1, b_0)$$

$$N = (n_{g-1}, n_{g-2}, \dots, n_1, n_0)$$

$$Y = (y_g, y_{g-1}, \dots, y_1, y_0)$$

$2^{-k} g$ は剰余の法Nの上での $2^k g$ の逆数であり、 n'_0 は、 $R = 2^k g$ として $R \cdot R^{-1} - N \cdot N' = 1$ ($0 \leq R^{-1} < N$, $0 \leq N' < R$)を満たす N' を k ビット＝1ブロックで分割した時の最下位ブロック（0番目のブロック）を表す。また、 c_1 、 c_2 、 m 、 tmp は k ビットのワーク変数であり、 i および j はループ変数である。

【0007】

図12のアルゴリズムを⑥の処理を除いて回路化した場合、図13のような回路になることが多いと考えられる。図13の回路では、A、B、およびNがそれぞれg個のブロックに分割されて、ブロック毎にブロック単位演算器101に入力され、演算結果 y'_i および y'_{i-1} がそれぞれYのブロック y_i および y_{i-1} に格納される。このとき、演算に必要な y_i がYから読み出されてブロック単位演算器101に入力される。

【0008】

ブロック単位演算器101は、図13の下方に示すように、レジスタ111、112、113、114、115、121、122、123、124、125、126、セクタ116、117、118、119、および乗算加算器120を備える。乗算加算器120は $A \times B + C + D$ の演算を行う。図13の回路と図12のアルゴリズムの①～⑤の処理の対応関係は次の通りである。

①の処理： $(c1, tmp) = a_i * b_j + y_i + c1$

セクタ116により $A = a_i$ と選択し、セクタ117により $B = b_j$ と選択し、セクタ118により $C = c1$ と選択し、セクタ119により $D = y_i$ と選択して、乗算加算器120の出力の上位kビットをc1としてレジスタ121に格納し、下位kビットをtmpとしてレジスタ126に格納する。

②の処理： $m = tmp * n'_0$

セクタ116により $A = tmp$ と選択し、セクタ117により $B = n'_0$ と選択し、セクタ118により $C = 0$ と選択し、セクタ119により $D = 0$ と選択して、乗算加算器120の出力の下位kビットをmとしてレジスタ125に格納する。

③の処理： $(c2, tmp) = m * n_i + tmp + c2$

セクタ116により $A = n_i$ と選択し、セクタ117により $B = m$ と選択し、セクタ118により $C = tmp$ と選択し、セクタ119により $D = c2$ と選択して、乗算加算器120の出力の上位kビットをc2としてレジスタ122に格納し、下位kビットをtmpとしてレジスタ126に格納する。

④の処理： $(c2, y_{i-1}) = m * n_i + tmp + c2$

セクタ 116 により $A = n_i$ と選択し、セクタ 117 により $B = m$ と選択し、セクタ 118 により $C = t_{mp}$ と選択し、セクタ 119 により $D = c_2$ と選択して、乗算加算器 120 の出力の上位 k ビットを c_2 としてレジスタ 122 に格納し、下位 k ビットを y'_{i-1} としてレジスタ 124 に格納する。レジスタ 124 の内容は Y の y_{i-1} に格納される。

⑤の処理: $(y_i, y_{i-1}) = y_i + c_1 + c_2$

セクタ 116 により $A = y'_i$ と選択し、セクタ 117 により $B = 1$ と選択し、セクタ 118 により $C = c_1$ と選択し、セクタ 119 により $D = c_2$ と選択して、乗算加算器 120 の出力の上位 k ビットを y'_i としてレジスタ 123 に格納し、下位 k ビットを y'_{i-1} としてレジスタ 124 に格納する。レジスタ 123 および 124 の内容はそれぞれ Y の y_i および y_{i-1} に格納される。

【0009】

図 13 の回路に図 12 のアルゴリズムの⑥の処理を追加して、モンゴメリ乗算剰余の多倍長演算アルゴリズムの回路として完成させる場合、⑥以外の処理と同じように、 Y および N を k ビット = 1 ブロックで分割してブロック毎の演算を繰り返す図 14 のような多倍長減算を採用して、図 15 のような回路にすることが多いと考えられる。

【0010】

図 15 では、図 14 の上方に示す多倍長減算を 2 の補数表現を用いてその下方に示すような加算に変形してから回路化している。2 の補数は、コンピュータ内で負数を表現するために用いられ、1 の補数に 1 を加えることで得られる。また、1 の補数は、ビットパターンを反転させることで得られる。

【0011】

図 15 では、図 13 のブロック単位演算器 101 がブロック単位演算器 131 に置き換えられている。ブロック単位演算器 131 は、ブロック単位演算器 101 に反転／非反転回路 141 を加えた構成を有する。

【0012】

図 15 の回路の動作のうち、図 12 のアルゴリズムの⑥以外の処理（通常処理

）に対応する動作は図 13 の回路と同じであるため、⑥の処理（最終処理）に対応する動作のみを以下に説明する。この処理では、最終処理状態信号により、反転／非反転回路 141 は n_i を反転して出力し、レジスタ 123 の y'_i は 1 に初期化され、ブロック単位演算器 131 から出力された y'_{i-1} は Y の y_i に格納される。

(1) y'_i を 1 に初期化する。

(2) 次の動作を $0 \leq i \leq g$ の範囲で繰り返す。

【0013】

セクタ 116 により $A = n_i$ の反転（1の補数）と選択し、セクタ 117 により $B = 1$ と選択し、セクタ 118 により $C = y'_{i-1} - n_{i-1}$ の結果からのキャリー）と選択し、セクタ 119 により $D = y_i$ と選択して、乗算加算器 120 の出力の上位 k ビットを y'_i としてレジスタ 123 に格納する。

(3) y'_i の bit_0 （最下位ビット）が 1（ $Y \geq N$ ）の場合は（4）へ進み、 y'_i の bit_0 が 0（ $Y < N$ ）の場合は（5）へ進む。

(4) 次の動作を $0 \leq i \leq g$ の範囲で繰り返す。

【0014】

セクタ 116 により $A = n_i$ の反転（1の補数）と選択し、セクタ 117 により $B = 1$ と選択し、セクタ 118 により $C = y'_{i-1} - n_{i-1}$ の結果からのキャリー）と選択し、セクタ 119 により $D = y_i$ と選択して、乗算加算器 120 の出力の上位 k ビットを y'_i としてレジスタ 123 に格納し、下位 k ビットを y'_{i-1} としてレジスタ 124 に格納する。レジスタ 124 の内容は Y の y_i に格納される。

(5) 動作を終了する

【0015】

【特許文献 1】

特開平 11-212456 号公報

【0016】

【発明が解決しようとする課題】

しかしながら、上述した仮想的な多倍長演算回路には、次のような問題がある。

【0017】

図15の回路において、A、B、N、Y、c1、c2、m、およびtmpを格納する回路は、通常、動作クロックに同期したRAM (random access memory) もしくはFF (flip-flop) で構成される。このため、乗算加算器120とその直前のセクタ116~119は、上記RAMもしくはFFの出力(A、B、N、Y、c1、c2、m、tmp)からRAMもしくはFFの入力(Y、c1、c2、m、tmp)までの通り道となるので、乗算加算器120とセクタ116~119の総遅延時間が動作クロックの周期より小さくしなければ、回路全体として動作しないということになる。

【0018】

したがって、図15の回路の動作周波数を向上させる際のボトルネックは、ブロック単位演算器131の最大遅延経路 $n_i \rightarrow$ 反転／非反転回路141 \rightarrow セクタ116 \rightarrow 乗算加算器120 \rightarrow (c1, c2, y'_i , y'_{i-1} , m, tmp) \rightarrow であり、この経路をいかに短縮するかが問題となる。

【0019】

本発明の課題は、モンゴメリ乗算剰余の多倍長演算を行う回路においてブロック単位演算器における減算のための遅延時間を短縮し、動作周波数を維持したままで多倍長演算を行うための演算装置を提供することである。

【0020】

【課題を解決するための手段】

図1は、本発明の演算装置の原理図である。図1の演算装置は、部分積生成手段201、エンコード手段202、選択手段203、および加算手段204を備え、ビットパターンで表された被乗数Aと乗数Bの乗算を行う。

【0021】

部分積生成手段201は、被乗数Aから2次Boothアルゴリズムにおける複数の部分積を生成する。エンコード手段202は、乗数Bを2次Boothアルゴリズムによりエンコードして、乗数Bの連続する3つのビットである b_{2i}

$+1$ 、 b_{2i} 、および b_{2i-1} を指定する i の値に応じた選択信号を出力する。選択手段 203 は、選択信号に応じて、複数の部分積のいずれかを選択して出力する。加算手段 204 は、選択手段 203 から出力される、 i の数と同じ数だけの部分積を加算して、乗算結果を生成する。

【0022】

そして、エンコーダ手段 202 が、 i が 0 のときには $-A$ を表す部分積を選択するための選択信号を出力し、 i が 0 以外のときには 0 を表す部分積を選択するための選択信号を出力し、加算手段 204 が、 $-A$ を表す部分積から被乗数 A の 2 の補数を生成して、被乗数 A の 2 の補数を乗算結果として出力するような動作モードを、演算装置に設ける。この動作モードにおいて、加算手段 204 が、 $-A$ を表す部分積から被乗数 A の 2 の補数を生成する代わりに、被乗数 A の 1 の補数を生成するようにすることもできる。

【0023】

2 次 Booth アルゴリズムは、 A および B が 2 の補数の整数である場合の乗算 $A \times B$ において部分積の個数を削減するためのアルゴリズムとして知られている。部分積生成手段 201 は、被乗数 A から 2 次 Booth アルゴリズムにおける複数の部分積として、 A 、 $2A$ 、 $-A$ 、 $-2A$ 、および 0 を表す部分積を生成する。

【0024】

上記動作モード（後述する補数モード）において、エンコーダ手段 202 は、乗数 B の連続する 3 つのビット b_{2i+1} 、 b_{2i} 、および b_{2i-1} をエンコードして、 b_1 、 b_0 、および b_{-1} の 3 つのビットに対しては $-A$ を表す部分積を選択するための選択信号を生成し、 b_{2i+1} 、 b_{2i} 、および b_{2i-1} ($i \neq 0$) の 3 つのビットに対しては 0 を表す部分積を選択するための選択信号を生成する。

【0025】

これにより、選択手段 203 は、 b_1 、 b_0 、および b_{-1} の 3 つのビットに対しては $-A$ を表す部分積を選択し、それ以外のビットに対しては 0 を表す部分積を選択するので、加算手段 204 は、 $-A$ を表す部分積と 0 とを加算すること

になる。 $-A$ は A の2の補数または A の1の補数で表されるから、この動作モードにおける乗算結果は、 A の2の補数または A の1の補数($=-A$)となる。

【0026】

これに対して、通常の動作モードにおいては、演算装置は乗算 $A \times B$ の結果を出力することができる。このように、乗算 $A \times B$ を行う演算装置において $-A$ を乗算結果として出力する動作モードを設けることで、図15の反転／非反転回路141がなくても図12の⑥の処理を行うことが可能となる。この演算装置をブロック単位演算器131に組み込めば、ブロック単位演算器131における遅延時間を短縮して、動作周波数を向上させることができる。

【0027】

図1の部分積生成手段201は、例えば、後述する図5のシフト回路241、243、および反転回路242に対応し、図1のエンコーダ手段202は、例えば、図5のエンコーダ247に対応する。また、図1の選択手段203は、例えば、図5の選択回路244に対応し、図1の加算手段204は、例えば、図5の多段加算回路245に対応する。

【0028】

【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。

本実施形態では、乗算加算器のアーキテクチャに2次Boothアルゴリズムを採用して、図15の回路のブロック単位演算器131の最大遅延経路 $\langle n_i \rightarrow$ 反転／非反転回路141 \rightarrow セレクタ116 \rightarrow 乗算加算器120 $\rightarrow (c_1, c_2, y'_i, y'_{i-1}, m, tmp) \rangle$ 上の反転／非反転回路141を乗算加算器120に吸収させる。これにより、ブロック単位演算器131における遅延時間を短縮して、動作周波数を向上させる。

【0029】

また、図15の回路では、反転／非反転回路141の規模がブロックビット数 k に比例するため、 k が大きくなるほど、図13の回路の規模に対する増分が顕著になる。しかし、本実施形態では、反転／非反転回路141を乗算加算器120に吸収することにより、図13の回路とほぼ同じ規模で図12のアルゴリズム

を実現することができる。

【0030】

2次Boothアルゴリズムは、 A および B が $-2^{q-1} \leq A, B \leq 2^{q-1} - 1$ で q ビットの2の補数の整数である場合の乗算 $A \times B$ について、 A および B のビット表現をそれぞれ $\{a_{q-1}, a_{q-2}, \dots, a_1, a_0\}$ および $\{b_{q-1}, b_{q-2}, \dots, b_1, b_0\}$ として、以下の手順により乗算 $A \times B$ の部分積の個数を $q/2$ に削減するアルゴリズムである。

- ・ B の多項式表現である式(1)を式(2)に変形する。
- ・ さらに、 d_i を -2 、 -1 、 0 、 $+1$ 、および $+2$ を取りうるビットと定義して、式(2)を式(3)で表す。

$$\begin{aligned} \text{式(1)} \quad & -b_{q-1} 2^{q-1} + b_{q-2} 2^{q-2} + \dots \\ & + b_1 2^1 + b_0 2^0 \end{aligned}$$

$$\begin{aligned} \text{式(2)} \quad & \sum \left((-2b_{2i+1} + b_{2i} + b_{2i-1}) 2^{2i} \right) \\ & \times 0 \leq i \leq q/2 - 1, \quad b_{-1} = 0 \end{aligned}$$

$$\begin{aligned} \text{式(3)} \quad & \sum d_i 2^{2i} \quad (\times d_i = -2b_{2i+1} + b_{2i} + b_{2i-1}) \\ & \times 0 \leq i \leq q/2 - 1, \quad b_{-1} = 0 \end{aligned}$$

q は偶数である必要があるが、 q が奇数であっても $+1$ 補正して2次Boothアルゴリズムを適用することができる。 A および B が $0 \leq A, B \leq 2^{k-1}$ の k ビットの整数である場合の乗算 $A \times B$ についても、 k が偶数であれば $q = k + 2$ 、 $a_{q-1} = a_{q-2} = b_{q-1} = b_{q-2}$ として、 k が奇数であれば $q = k + 1$ 、 $a_{q-1} = b_{q-1}$ として2次Boothアルゴリズムを適用することができる。

【0031】

図2は、 A および B が $0 \leq A, B \leq 2^{k-1}$ の k ビットの整数である場合の2次Boothアルゴリズムによる乗算 $A \times B$ を示している。この場合の動作は次

の通りである。

- ① k ビットの A および B を q ビットに補正して A' および B' とする。このとき、 k が偶数であれば $q = k + 2$ 、 $a_{q-1} = a_{q-2} = b_{q-1} = b_{q-2}$ とし、 k が奇数であれば $q = k + 1$ 、 $a_{q-1} = b_{q-1}$ とする。
- ② $0 \leq i \leq q/2 - 1$ とした場合の B' の 3 つのビット b_{2i+1} 、 b_{2i} 、および b_{2i-1} を、Booth エンコーダ 211 を用いて d_i にエンコードする。
- ③ b_{2i+1} 、 b_{2i} 、および b_{2i-1} から得られた d_i を -2 、 -1 、 0 、 $+1$ 、および $+2$ を取りうる B' のビットとして用いて、乗算 $A' \times B'$ を行う。この乗算における各部分積は $A' \times d_i$ (2 の補数) である。
- ④ 積 $A' \times B'$ ($2q$ ビット) から下位 $2k$ ビットを切り出し、積 $A \times B$ ($2k$ ビット) とする。

【0032】

図3は、2次Boothアルゴリズムを採用した乗算器に、出力が $-A$ (2の補数) になるモード (補数モード) を追加した乗算器を示している。図3の乗算器においては、モード選択信号により、通常の乗算 $A \times B$ を行うモード (通常モード) と補数モードとが切り替えられる。

【0033】

補数モード時には、Booth エンコーダ 221 の出力 d_0 が -1 になり、それ以外の出力 d_i ($i \neq 0$) が 0 になるため、部分積 $A' \times d_0$ は $-A'$ (2の補数) になり、他の部分積 $A' \times d_i$ ($i \neq 0$) は 0 になる。したがって、乗算器の出力 X は $-A$ (2の補数) になる。補数モード時の B は $don't\ care$ で構わない。

【0034】

ここでは、部分積 $A' \times d_0$ を 2 の補数としている。2 の補数は 1 の補数に 1 を加えたものであるから、2 の補数を回路で実現するときは、ビットパターンを 1 の補数にして (つまり、ビット反転して) から、 $+1$ 加算する。しかし、この操作をそのまま行くと、部分積加算にとっても時間がかかる。

【0035】

そこで、部分積 $A' \times d_0$ を 1 の補数としておき、2 の補数にするための +1 加算を部分積加算と一緒に行うのが一般的である。図 3 の Booth エンコーダ 221 の s_i は、この +1 加算を実現するための補正值として用いられ、 $d_i < 0$ のとき $s_i = 1$ となり、 $d_i \geq 0$ のとき $s_i = 0$ となる。

【0036】

図 3 の乗算器では、補数モード時の B が don't care で構わないのに対して、補数モード時に $B = 0$ を入力すれば、補数モード時の d_i ($i \neq 0$) として通常モード時と同じ値を用いることができる。

【0037】

図 4 は、このような乗算器を示している。補数モード時には、Booth エンコーダ 231 の出力 d_0 が -1 になり、それ以外の出力 d_i ($i \neq 0$) は、 $B = 0$ の場合の通常モード時の出力 d_i ($i \neq 0$) と同じ 0 になる。このため、部分積 $A' \times d_0$ は $-A'$ (2 の補数) になり、他の部分積 $A' \times d_i$ ($i \neq 0$) は 0 になる。したがって、乗算器の出力 X は $-A$ (2 の補数) になる。

【0038】

図 4 の乗算器は、図 3 の乗算器に比べて Booth エンコーダの規模が小さくて済むため、乗算器を使用する側で $B = 0$ を入力するのが容易である場合は、最適な乗算器であるといえる。

【0039】

図 3 および図 4 の乗算器は、例えば、図 5 に示すような回路で構成される。図 5 の回路は、レジスタ 240、246、シフト回路 241、243、反転回路 242、選択回路 244、多段加算回路 245、およびエンコーダ 247 を備える。エンコーダ 247 は、図 3 の Booth エンコーダ 221 または図 4 の Booth エンコーダ 231 に対応する。

【0040】

レジスタ 246 の B' は 2 ビットずつの組に分割され、各組のビット対が下位の組の上位ビットとともにエンコーダ 247 に順番に入力される。エンコーダ 247 は、これらの 3 つの入力 b_{2i+1} 、 b_{2i} 、および b_{2i-1} から d_i に相当する選択信号を生成し、 A' の倍数を選択する選択回路 244 に出力する。

【0041】

シフト回路241、243は、入力されたビットパターンを1ビットシフトし、反転回路242は、入力されたビットパターンを反転させる。これらの回路により、レジスタ240の A' からは、直接(A')、1ビットシフト($2A'$)、反転($-A'$ (1の補数))、および反転の1ビットシフト($-2A'$)の4種類の値が生成され、選択回路244に入力される。

【0042】

選択回路244は、エンコーダ247の各ビット対出力に対応して設けられており、エンコーダ247からの選択信号に応じて4種類の入力値の1つを選択し、部分積 $A' \times d_i$ として多段加算回路245に出力する。このとき、選択回路244からは補正值 s_i も同時に出力される。多段加算回路245は、入力された部分積 $A' \times d_i$ および補正值 s_i を加算して、乗算結果 $X (=A \times B)$ を出力する。

【0043】

図3および図4の乗算器では、補数モード時に $d_0 = -1$ および $s_0 = 1$ とすることで、出力を $-A$ (2の補数)としているが、 $s_0 = 0$ とすればこの出力を $-A$ (1の補数)にすることができる。図14に示したように、多倍長減算では、ビットパターン全体として2の補数であってもブロック単位の演算では1の補数を扱うことになるので、乗算器の出力を $-A$ (1の補数)にすることが重要である。この場合、部分積 $A' \times d_0$ に対応する $-A'$ を生成する回路は、図6に示すように、反転回路251および加算回路252を用いて構成される。

【0044】

図6の反転回路251は、 A' を反転して出力し、加算回路252は、反転された A' と補正值253とを加算して出力する。補正值253は、通常モード時に1となり、補数モード時に0となるので、通常モード時は $-A'$ (2の補数)が出力され、補数モード時は $-A'$ (1の補数)が出力される。

【0045】

図6の反転回路251および加算回路252は、それぞれ、図5の反転回路242および多段加算回路245に対応し、補正值253は補正值 s_0 に対応する

。この構成では、通常モードと補数モードとで s_0 が異なる値をとることになる。

【0046】

図5の乗算器の出力と、 $0 \leq C \leq 2^k - 1 - 1$ および $0 \leq D \leq 2^k - 1 - 1$ である k ビットの C および D とを加算する、 $A \times B + C + D$ の乗算加算器は、図7のような回路で構成される。図7の回路では、図5の多段加算回路245の代わりに、多段加算回路261が用いられている。

【0047】

多段加算回路261は、選択回路244からの部分積および補正值に加えて、外部から入力される C および D を加算し、乗算加算結果を $X (= A \times B + C + D)$ を出力する。補数モード時は、 $X = -A$ (2の補数) $+ C + D$ となる。

【0048】

しかし、図6の $-A'$ 生成回路のような構成を採用すれば、補数モード時の出力は、 $X = -A$ (1の補数) $+ C + D$ となる。この場合、図6の反転回路251および加算回路252は、それぞれ、図7の反転回路242および多段加算回路261に対応し、補正值253は補正值 s_0 に対応する。

【0049】

図8および図9は、図7において補数モード時の出力を $-A$ (1の補数) $+ C + D$ とした乗算加算器を使用して、図14の多倍長減算 ($Y = Y - N$) を実行する回路を示している。図8の多倍長減算回路は、レジスタ271、272、273、274、275、277、乗算加算器276、および反転回路278を備え、図9の多倍長減算回路は、図8において乗算加算器276の代わりに乗算加算器281を用いた構成を有する。

【0050】

図8の乗算加算器276は、図3のBoothエンコーダ221に従って乗算を行うため、入力 B は $don't\ care$ となっている。これに対して、図9の乗算加算器281は、図4のBoothエンコーダ231に従って乗算を行うため、入力 B は0となっている。図8および図9の回路の動作は次の通りである。

- (0) レジスタ 277 のキャリー (carry) の値を 1 に初期化する。
- (1) レジスタ 271 の N から n_i を選択してレジスタ 273 に格納し、レジスタ 272 の Y から y_i を選択してレジスタ 274 に格納する。
- (2) 乗算加算器 276 は、入力を $A = n_i$ 、 $B = \text{don't care}$ 、 $C[k-1:1] = 0$ 、 $C[0] = \text{carry}$ ($i-1$ 番目のブロックの演算結果からのキャリー)、 $D = y_i$ として、補数モード時の k ビットの演算 $X = -A$ (1 の補数) $+ C + D$ を実行する。

【0051】

また、乗算加算器 281 は、入力を $A = n_i$ 、 $B = 0$ 、 $C[k-1:1] = 0$ 、 $C[0] = \text{carry}$ 、 $D = y_i$ として、 $X = -A$ (1 の補数) $+ C + D$ を実行する。

(3) 乗算加算器 276 および 281 の出力のうち、 $X[k-1:0]$ を y'_{i-1} としてレジスタ 275 に保存し、 $X[k]$ を反転回路 278 により反転させてレジスタ 277 に格納する。レジスタ 277 に格納された carry は、 $i+1$ 番目のブロックへのキャリーとなる。ここで、 $X[k]$ を反転させるのは、図 14 に示したように、補数のキャリーが必要となるためである。 $X[2k-1:k+1]$ は不要なので、無視される。

- (4) レジスタ 275 の y'_{i-1} を y_i としてレジスタ 272 に格納する。
- (5) (1) ~ (4) の動作を $0 \leq i \leq g$ の範囲で繰り返す。
- (6) (5) の動作が終了した時点で、レジスタ 272 の Y には $Y-N$ の演算結果が格納されている。また、 g 番目のブロックの演算結果 $X[k]$ は、 $Y = Y - N$ の符号 ($1: Y \geq N$, $0: Y < N$) を表す。

【0052】

図 10 および図 11 は、図 7 において補数モード時の出力を $-A$ (1 の補数) $+ C + D$ とした乗算加算器を使用して、モンゴメリ乗算剰余の多倍長演算 ($Y = AB2^{-kg} \bmod N$) を実行する回路を示している。図 10 の多倍長演算回路は、レジスタ 291、292、295、299、ブロック選択回路 293、294、296、298、300、およびブロック単位演算器 297 を備え、図 11 の多倍長演算回路は、図 10 においてブロック単位演算器 297 の代わりに

ブロック単位演算器 331 を用いた構成を有する。

【0053】

図10において、レジスタ 291、292、および 295 には、A、B、および N がそれぞれ g 個のブロックに分割されて格納され、ブロック選択回路 293、294、および 296 は、それぞれ A、B、および N のブロック a_i 、 b_i 、および n_i を選択して、ブロック単位演算器 297 に出力する。また、ブロック選択回路 298 は、レジスタ 299 から Y のブロック y_i を選択して、ブロック単位演算器 297 に出力する。

【0054】

ブロック単位演算器 297 は、ブロック a_i 、 b_i 、 n_i 、 n'_0 、および y'_i を入力として演算を行い、演算結果 y'_i および y'_{i-1} を出力する。ブロック選択回路 300 は、 y'_i および y'_{i-1} のいずれかを選択し、ブロック y_i または y_{i-1} としてレジスタ 299 に格納する。図11の多倍長演算回路の動作も同様である。

【0055】

ブロック単位演算器 297 は、図10の下方に示すように、レジスタ 311、312、313、314、315、321、322、323、324、325、326、セクタ 316、317、318、319、乗算加算器 276、および反転回路 320 を備える。図11のブロック単位演算器 331 は、図10において乗算加算器 276 の代わりに乗算加算器 281 を用い、さらにセクタ 332 を追加した構成を有する。

【0056】

図10および図11の回路では、図12のアルゴリズムの⑥以外の処理（通常処理）と⑥の処理（最終処理）が、処理状態信号 301 によって切り替えられる。乗算加算器 276 および 281 は、通常処理のときは通常モードの演算（ $X = A \times B + C + D$ ）を行い、最終処理のときは補数モードの演算（ $X = -A$ （1の補数） $+ C + D$ ）を行う。また、図11のセクタ 332 は、通常処理のときは入力 1 を選択し、最終処理のときは入力 0 を選択して、乗算加算器 281 に出力する。

【0057】

図10および図11の回路の通常処理に対応する動作は図13の回路と同じであるため、最終処理に対応する動作のみを説明することにする。図10および図11の回路の処理⑥に相当する動作は次の通りである。

(0) レジスタ323の y'_i の値を1に初期化し、乗算加算器276および281を補数モードに設定する。

(1) レジスタ295のNから n_i を選択してレジスタ312に格納し、レジスタ299のYから y_i を選択してレジスタ315に格納する。

(2) セレクタ316、318、および319は、それぞれ、 $A=n_i$ 、 $C=y'_i$ のbit0を反転回路320により反転した値($i-1$ 番目のブロックの演算結果からのキャリー)、および $D=y_i$ と選択する。

【0058】

このとき、図10の回路では、セレクタ317の出力Bはdon't careで構わない。これに対して、図11の回路では、セレクタ332が入力0を選択し、セレクタ317がセレクタ332の出力を選択するので、結局、 $B=0$ と選択される。

【0059】

そして、乗算加算器276および281は、補数モード時のkビットの演算 $X=-A$ (1の補数) $+C+D$ を実行する。

(3) 乗算加算器276および281の出力のうち、 $X[k-1:0]$ を y'_{i-1} としてレジスタ324に保存し、 $X[2k-1:k]$ を y'_i としてレジスタ323に格納する。レジスタ323に格納された y'_i のbit0($X[k]$)は、反転回路320により反転されて、 $i+1$ 番目のブロックへのキャリーとなる。

(4) レジスタ324の y'_{i-1} を y_i としてレジスタ299に格納する。レジスタ323の y'_i のbit0以外のビットは不要なので、無視される。

(5) (1)～(4)の動作を $0 \leq i \leq g$ の範囲で繰り返す。

(6) (5)の動作が終了した時点で、レジスタ299のYには $Y-N$ の演算結果が格納されている。また、レジスタ323に格納された y'_g のbit0は、

$Y = Y - N$ の符号 ($1 : Y \geq N$, $0 : Y < N$) を表す。

【0060】

(付記1) ビットパターンで表された被乗数Aと乗数Bの乗算を行う演算装置であって、

被乗数Aから2次Boothアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数Bを前記2次Boothアルゴリズムによりエンコードして、乗数Bの連続する3つのビットである b_{2i+1} 、 b_{2i} 、および b_{2i-1} を指定する i の値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、 i の数と同じ数だけの部分積を加算して、乗算結果を生成する加算手段とを備え、

前記エンコーダ手段が、 i が0のときには $-A$ を表す部分積を選択するための選択信号を出力し、 i が0以外のときには0を表す部分積を選択するための選択信号を出力し、前記加算手段が、該 $-A$ を表す部分積から被乗数Aの2の補数を生成して、該被乗数Aの2の補数を前記乗算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【0061】

(付記2) ビットパターンで表された被乗数Aと乗数Bの乗算を行う演算装置であって、

被乗数Aから2次Boothアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数Bを前記2次Boothアルゴリズムによりエンコードして、乗数Bの連続する3つのビットである b_{2i+1} 、 b_{2i} 、および b_{2i-1} を指定する i の値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、 i の数と同じ数だけの部分積を加算して、乗算

結果を生成する加算手段とを備え、

前記エンコーダ手段が、 i が 0 のときには $-A$ を表す部分積を選択するための選択信号を出力し、 i が 0 以外のときには 0 を表す部分積を選択するための選択信号を出力し、前記加算手段が、該 $-A$ を表す部分積から被乗数 A の 1 の補数を生成して、該被乗数 A の 1 の補数を前記乗算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【0062】

(付記 3) 前記エンコーダ手段は、前記動作モードにおいて、乗数 B の値にかかわらず、 i が 0 以外のときには 0 を表す部分積を選択するための選択信号を出力することを特徴とする付記 2 記載の演算装置。

【0063】

(付記 4) 前記エンコーダ手段は、前記動作モードにおいて、乗数 B として 0 が入力された場合に、 i が 0 以外のときには 0 を表す部分積を選択するための選択信号を出力することを特徴とする付記 2 記載の演算装置。

【0064】

(付記 5) ビットパターンで表された被乗数 A と乗数 B を乗算し、乗算結果とビットパターンで表された数 C および数 D とを加算する、乗算加算を行う演算装置であって、

被乗数 A から 2 次 Booth アルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数 B を前記 2 次 Booth アルゴリズムによりエンコードして、乗数 B の連続する 3 つのビットである b_{2i+1} 、 b_{2i} 、および b_{2i-1} を指定する i の値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、 i の数と同じ数だけの部分積と数 C と数 D とを加算して、乗算加算結果を生成する加算手段とを備え、

前記エンコーダ手段が、 i が 0 のときには $-A$ を表す部分積を選択するための選択信号を出力し、 i が 0 以外のときには 0 を表す部分積を選択するための選択

信号を出力し、前記加算手段が、該 $-A$ を表す部分積から被乗数 A の1の補数を生成し、該被乗数 A の1の補数と数 C と数 D とを加算した結果を前記乗算加算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【0065】

(付記6) k ビットのビットパターンで表された被乗数 A と乗数 B を乗算し、乗算結果と k ビットのビットパターンで表された数 C および数 D とを加算することで、1ブロックが k ビットである $g+1$ 個のブロックからなる整数 Y から1ブロックが k ビットである g 個のブロックからなる整数 N を減算する演算を行う演算装置であって、

被乗数 A から2次Boothアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数 B を前記2次Boothアルゴリズムによりエンコードして、乗数 B の連続する3つのビットである b_{2i+1} 、 b_{2i} 、および b_{2i-1} を指定する i の値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、 i の数と同じ数だけの部分積と数 C と数 D とを加算して、 $2k$ ビットの乗算加算結果を生成する加算手段と、

前記乗算加算結果の一部のビットを反転する反転手段とを備え、

前記部分積生成手段が、整数 N の j 番目のブロック n_j を被乗数 A として用い、前記エンコーダ手段が、 i が0のときには $-A$ を表す部分積を選択するための選択信号を出力し、 i が0以外のときには0を表す部分積を選択するための選択信号を出力し、前記加算手段が、該 $-A$ を表す部分積から被乗数 A の1の補数を生成し、 $j-1$ 番目のブロックの乗算加算からのキャリーを数 C として用い、整数 Y の j 番目のブロック y_j を数 D として用いて、該被乗数 A の1の補数と数 C と数 D とを加算した結果を j 番目のブロックの乗算加算結果として出力し、前記反転手段が、該 j 番目のブロックの乗算加算結果の一部のビットを反転して $j+1$ 番目のブロックの乗算加算へのキャリーを生成することを特徴とする演算装置。

。

【0066】

(付記7) ビットパターンで表された整数IおよびJと剰余の法Nを1ブロックがkビットであるg個のブロックにそれぞれ分割して、 $Y = I J 2^{-k g} \bmod N$ となるモンゴメリ乗算剰余の多倍長演算を行う演算装置であって、

kビットの被乗数A、乗数B、数C、および数Dのそれぞれについて、与えられた複数の値のいずれかを選択して出力する第1の選択手段と、

前記第1の選択手段から出力される被乗数Aから2次Boothアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

前記第1の選択手段から出力される乗数Bを前記2次Boothアルゴリズムによりエンコードして、乗数Bの連続する3つのビットである b_{2i+1} 、 b_{2i} 、および b_{2i-1} を指定するiの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する第2の選択手段と、

前記第2の選択手段から出力される、iの数と同じ数だけの部分積と、前記第1の選択手段から出力される数Cと数Dとを加算して、2kビットの乗算加算結果を生成する加算手段と、

前記乗算加算結果の一部のビットを反転する反転手段とを備え、

前記第1の選択手段が、整数Nのj番目のブロック n_j を被乗数Aとして選択し、j-1番目のブロックの乗算加算からのキャリーを数Cとして選択し、Yのj番目のブロック y_j を数Dとして選択し、前記エンコーダ手段が、iが0のときには-Aを表す部分積を選択するための選択信号を出力し、iが0以外のときには0を表す部分積を選択するための選択信号を出力し、前記加算手段が、該-Aを表す部分積から被乗数Aの1の補数を生成し、該被乗数Aの1の補数と数Cと数Dとを加算した結果をj番目のブロックの乗算加算結果として出力し、前記反転手段が、該j番目のブロックの乗算加算結果の一部のビットを反転してj+1番目のブロックの乗算加算へのキャリーを生成するような動作モードを設けたことを特徴とする演算装置。

【0067】

【発明の効果】

本発明によれば、モンゴメリ乗算剰余の多倍長減算を行うために必要となる回路を乗算加算器に吸収させることができ、モンゴメリ乗算剰余の多倍長演算アルゴリズムを、従来の回路より小さい規模のハードウェアで実現することが可能となる。

【0068】

また、多倍長減算を行うために必要となる回路を乗算加算器に吸収させることで、回路の遅延時間が短縮されるため、従来の回路より高い動作周波数で動作させることができる。

【0069】

モンゴメリ乗算剰余の多倍長演算アルゴリズムは、R S A暗号や楕円曲線暗号で使われる演算の中では計算量が多いものであり、本発明はこれらの暗号処理の速度向上に寄与するところが多い。

【図面の簡単な説明】**【図1】**

本発明の演算装置の原理図である。

【図2】

2次B o o t hアルゴリズムによる演算を示す図である。

【図3】

第1の乗算器を示す図である。

【図4】

第2の乗算器を示す図である。

【図5】

乗算器の回路構成図である。

【図6】

- A' 生成回路を示す図である。

【図7】

乗算加算器の回路構成図である。

【図8】

第 1 の多倍長減算回路を示す図である。

【図 9】

第 2 の多倍長減算回路を示す図である。

【図 1 0】

第 1 の多倍長演算回路を示す図である。

【図 1 1】

第 2 の多倍長演算回路を示す図である。

【図 1 2】

モンゴメリ乗算剰余の多倍長演算アルゴリズムを示す図である。

【図 1 3】

仮想的な乗算剰余演算回路を示す図である。

【図 1 4】

多倍長減算を示す図である。

【図 1 5】

仮想的な多倍長演算回路を示す図である。

【符号の説明】

1 0 1、1 3 1、2 9 7、3 3 1 ブロック単位演算器
1 1 1、1 1 2、1 1 3、1 1 4、1 1 5、1 2 1、1 2 2、1 2 3、1 2 4
、1 2 5、1 2 6、2 4 0、2 4 6、2 7 1、2 7 2、2 7 3、2 7 4、2 7 5
、2 7 7、2 9 1、2 9 2、2 9 5、2 9 9、3 1 1、3 1 2、3 1 3、3 1 4
、3 1 5、3 2 1、3 2 2、3 2 3、3 2 4、3 2 5、3 2 6、 レジスタ
1 1 6、1 1 7、1 1 8、1 1 9、3 1 6、3 1 7、3 1 8、3 1 9、3 3 2
セレクタ
1 4 1 反転／非反転回路
1 2 0、2 7 6、2 8 1 乗算加算器
2 0 1 部分積生成手段
2 0 2 エンコーダ手段
2 0 3 選択手段
2 0 4 加算手段

2 1 1、2 2 1、2 3 1 B o o t h エンコーダ

2 4 1、2 4 3 シフト回路

2 4 2、2 5 1、2 7 8、3 2 0 反転回路

2 4 4 選択回路

2 4 5、2 6 1 多段加算回路

2 4 7 エンコーダ

2 5 2 加算回路

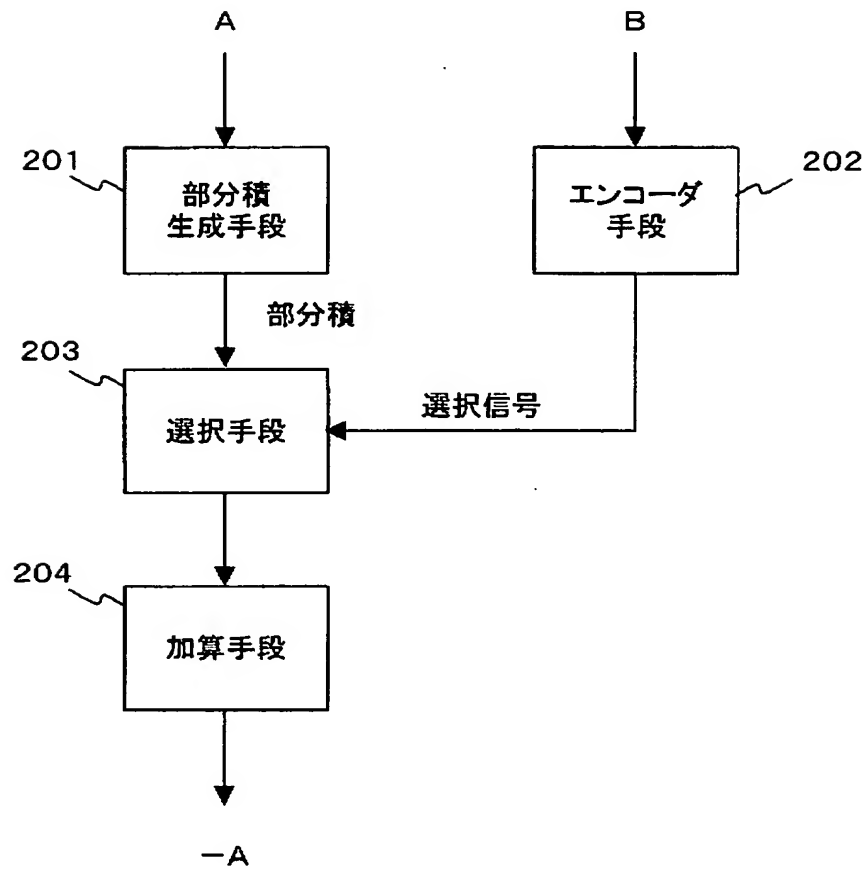
2 5 3 補正值

2 9 3、2 9 4、2 9 6、2 9 8、3 0 0 ブロック選択回路

【書類名】 図面

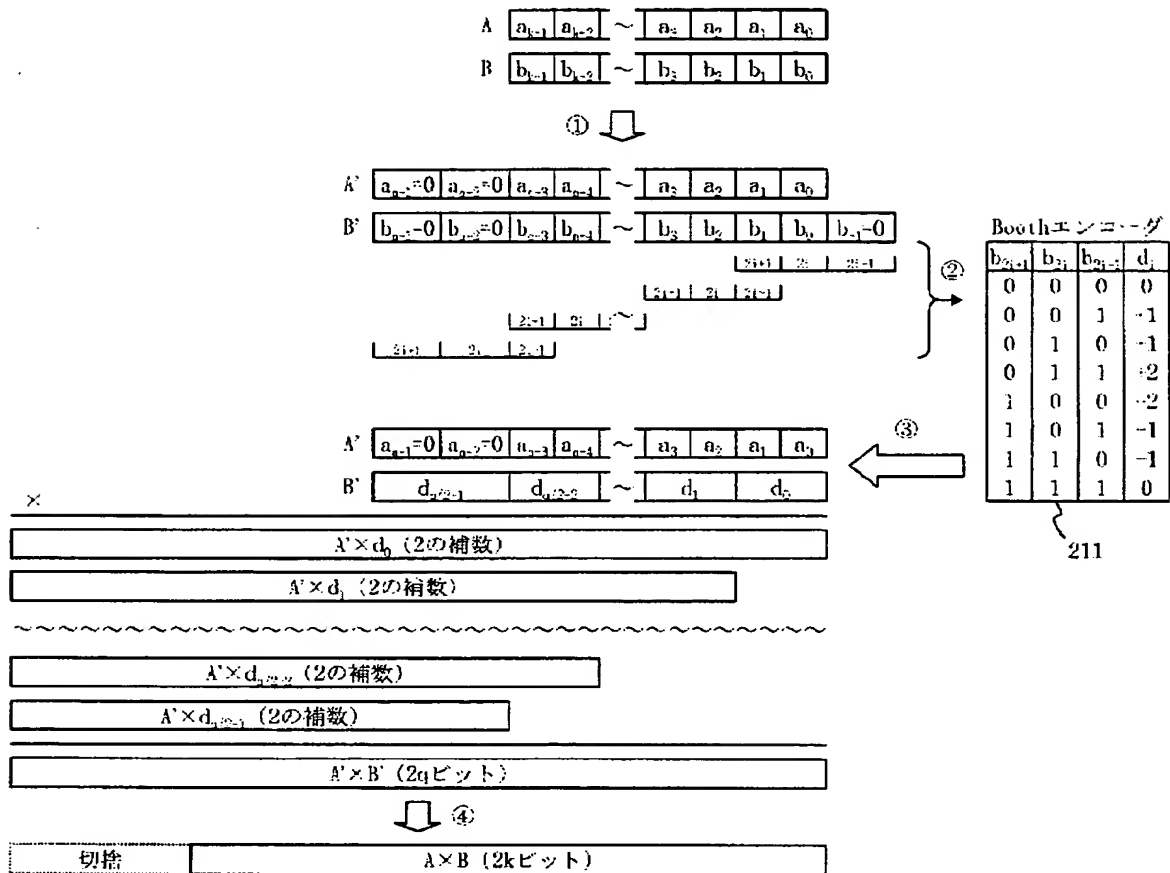
【図 1】

本発明の演算装置の原理図



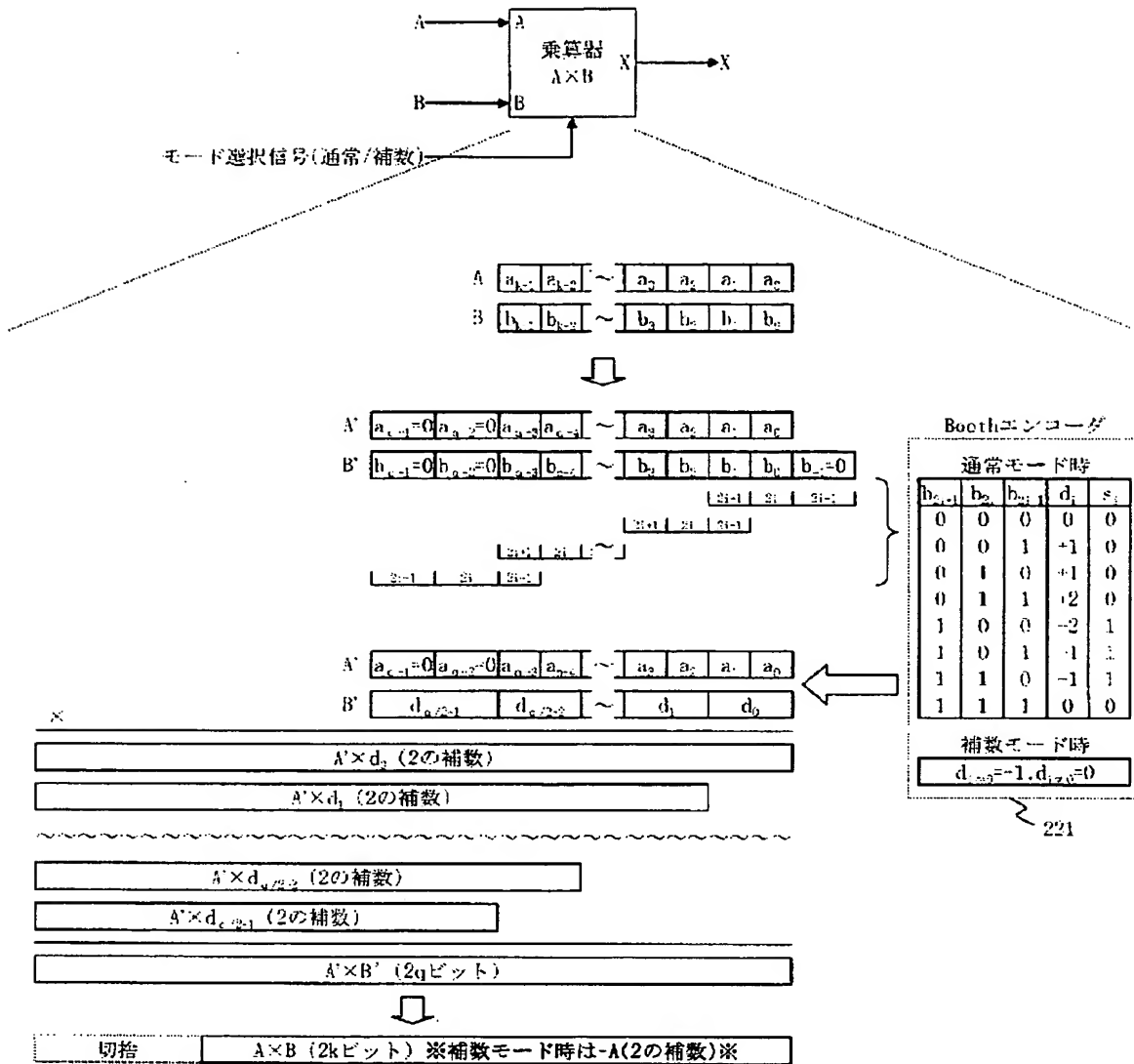
【図 2】

2次Boothアルゴリズムによる乗算を示す図



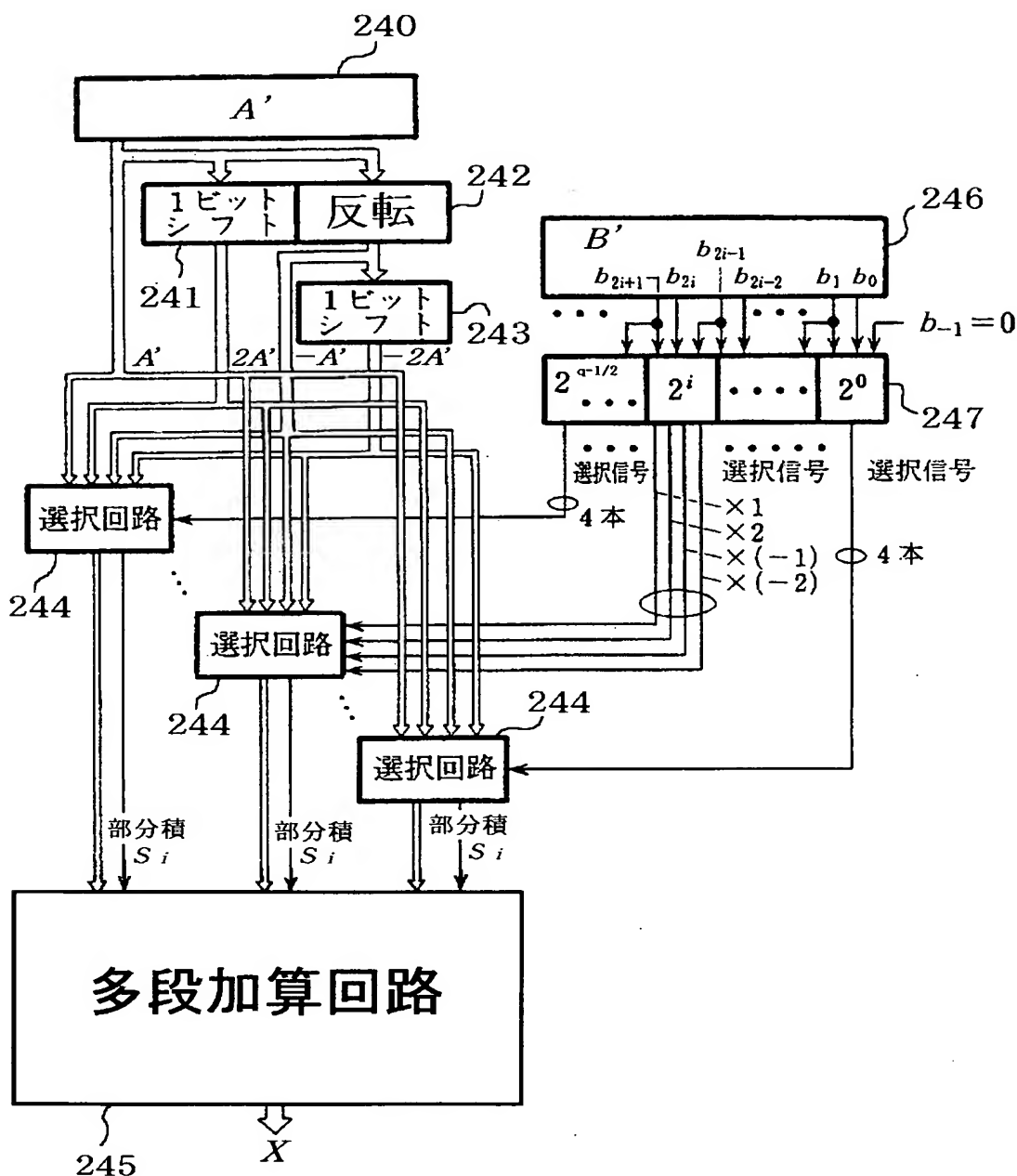
【図 3】

第1の乗算器を示す図



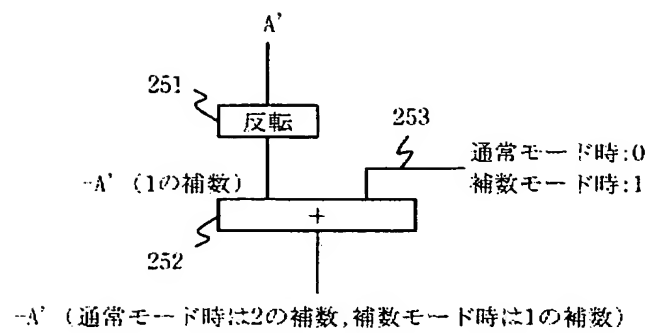
【図 5】

乗算器の回路構成図



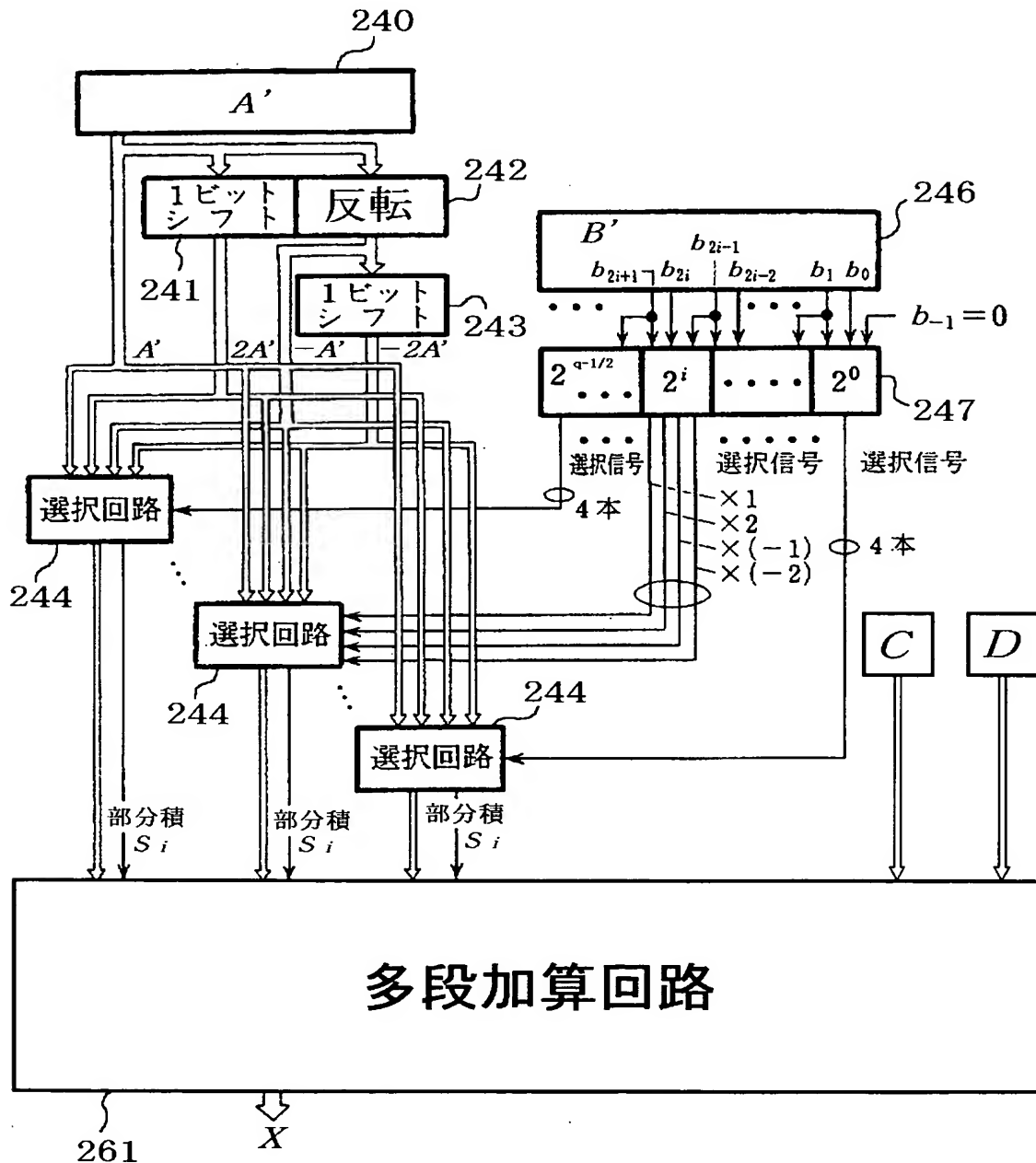
【図 6】

-A' 生成回路を示す図



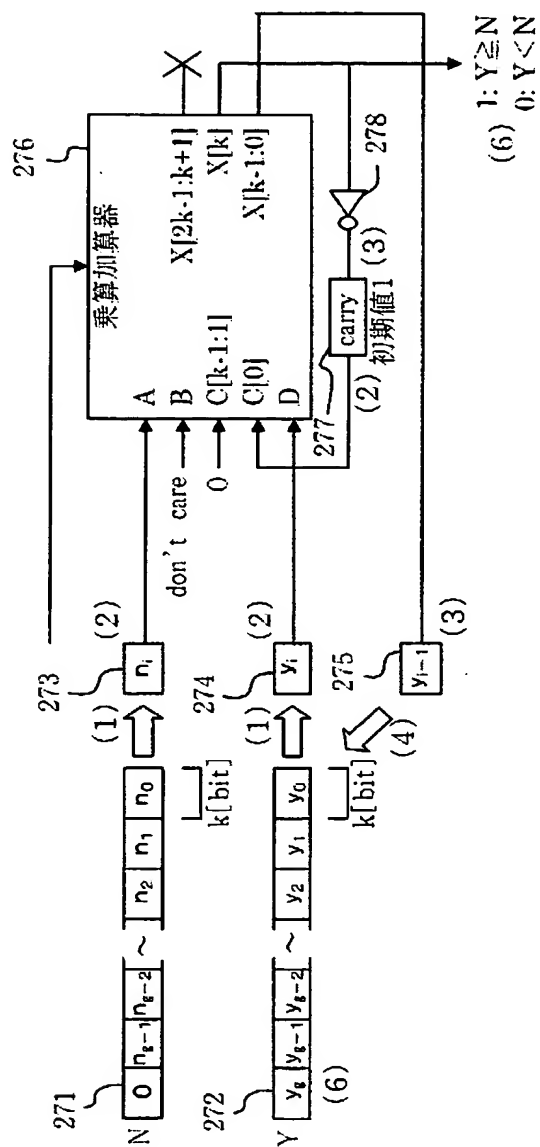
【図 7】

乗算加算器の回路構成図



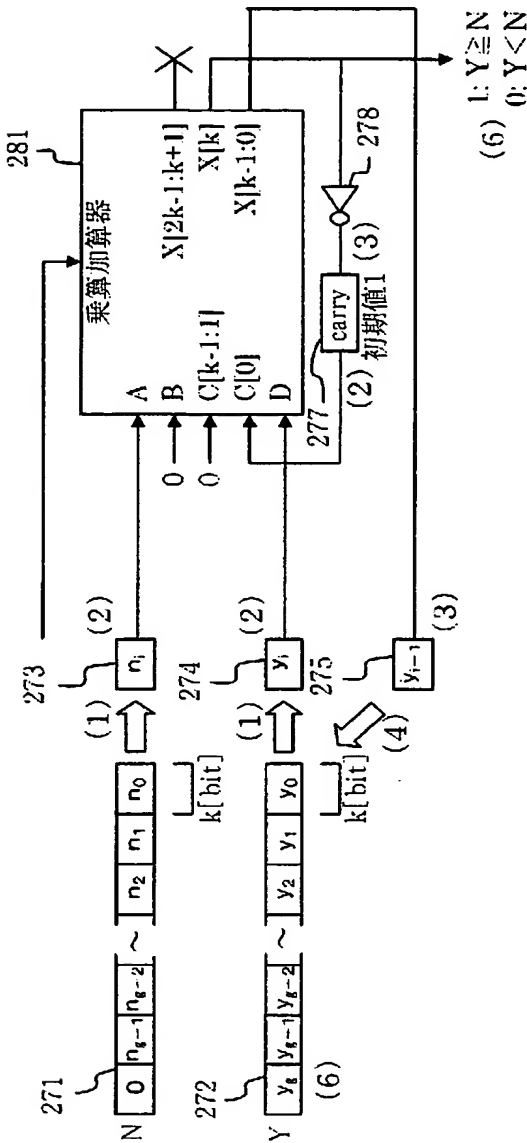
【図 8】

第1の多倍長減算回路を示す図



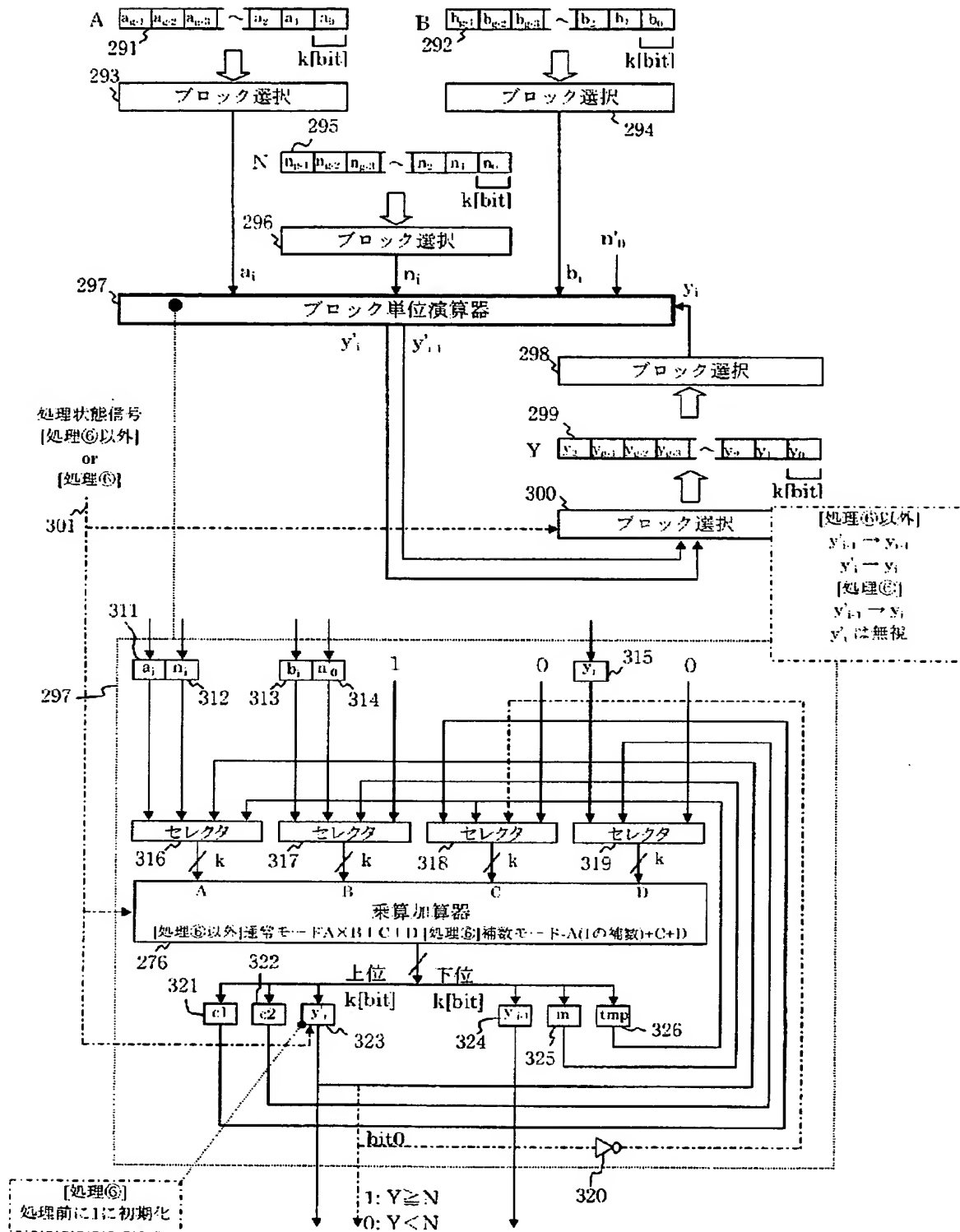
【図 9】

第2の多倍長減算回路を示す図



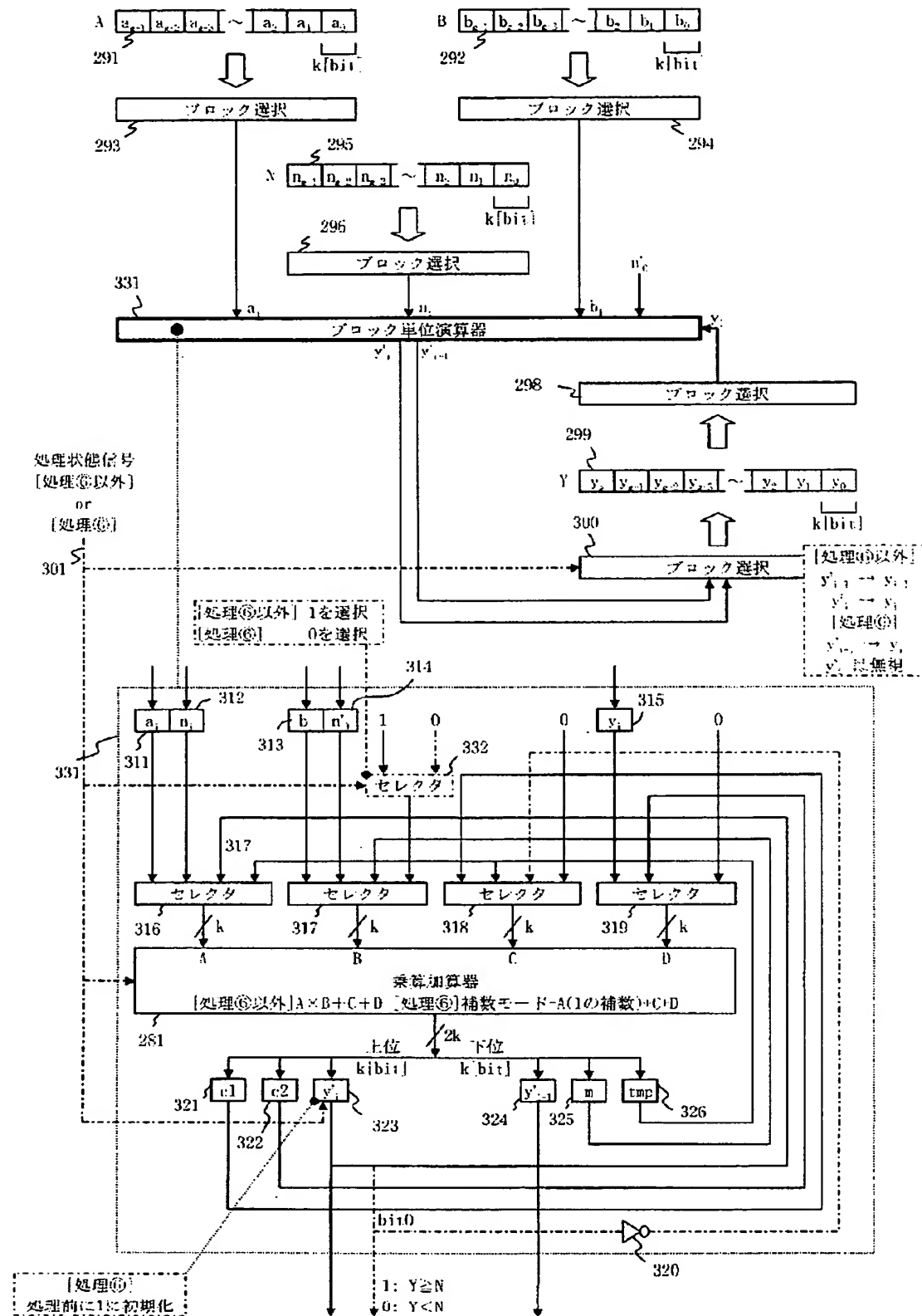
【図 10】

第1の多倍長演算回路を示す図



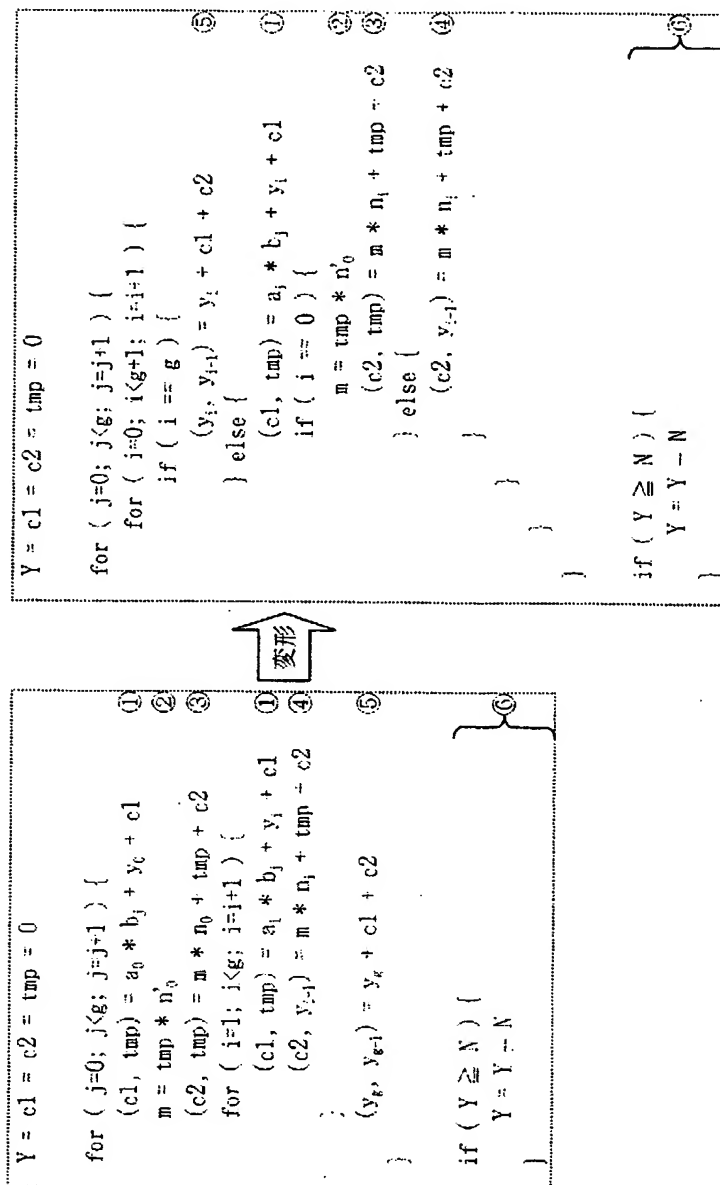
【図 11】

第2の多倍長演算回路を示す図



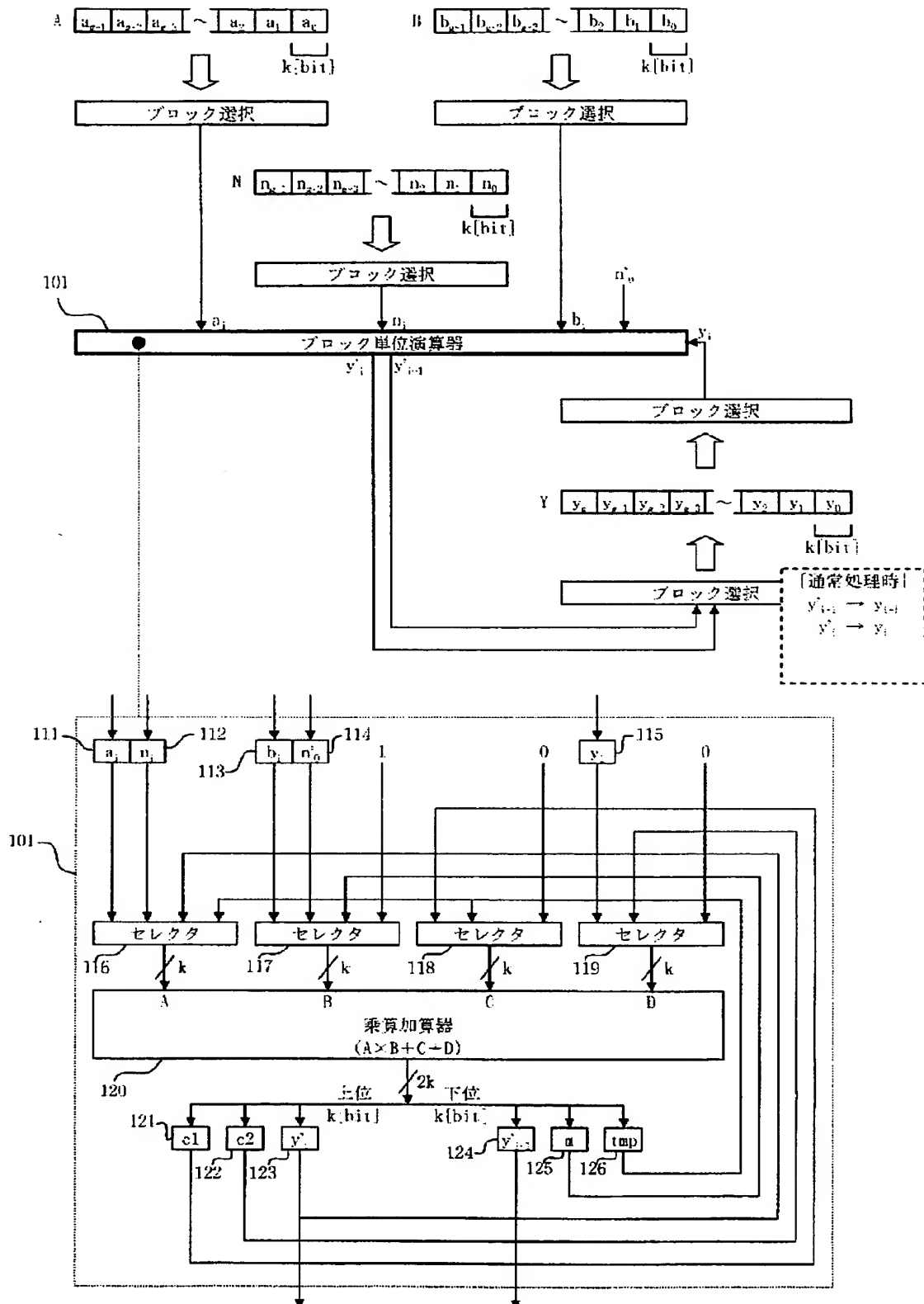
【図 12】

モンゴメリ乗算剰余の多倍長演算アルゴリズムを示す図



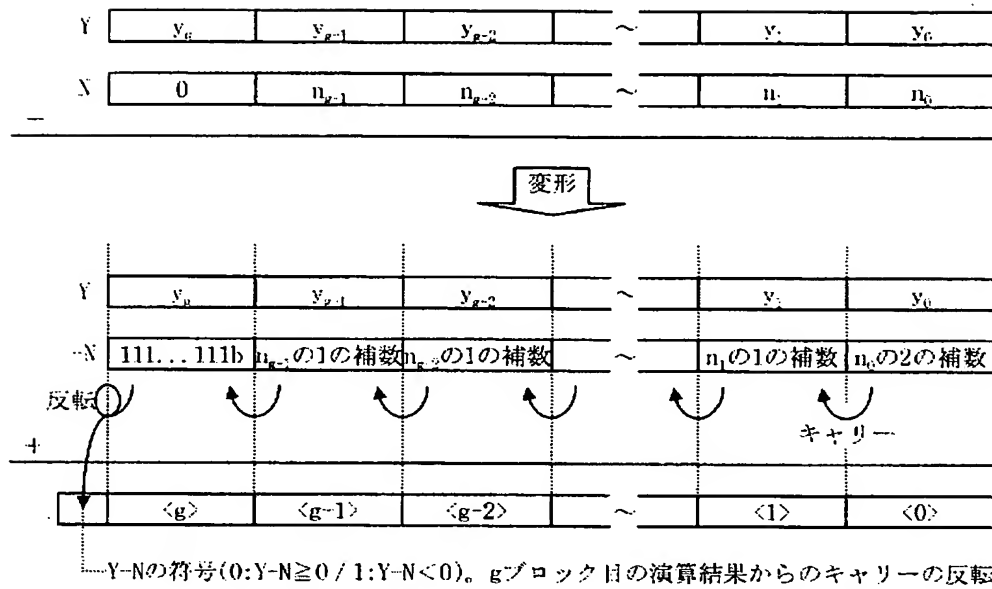
【図 13】

仮想的な乗算剰余演算回路を示す図



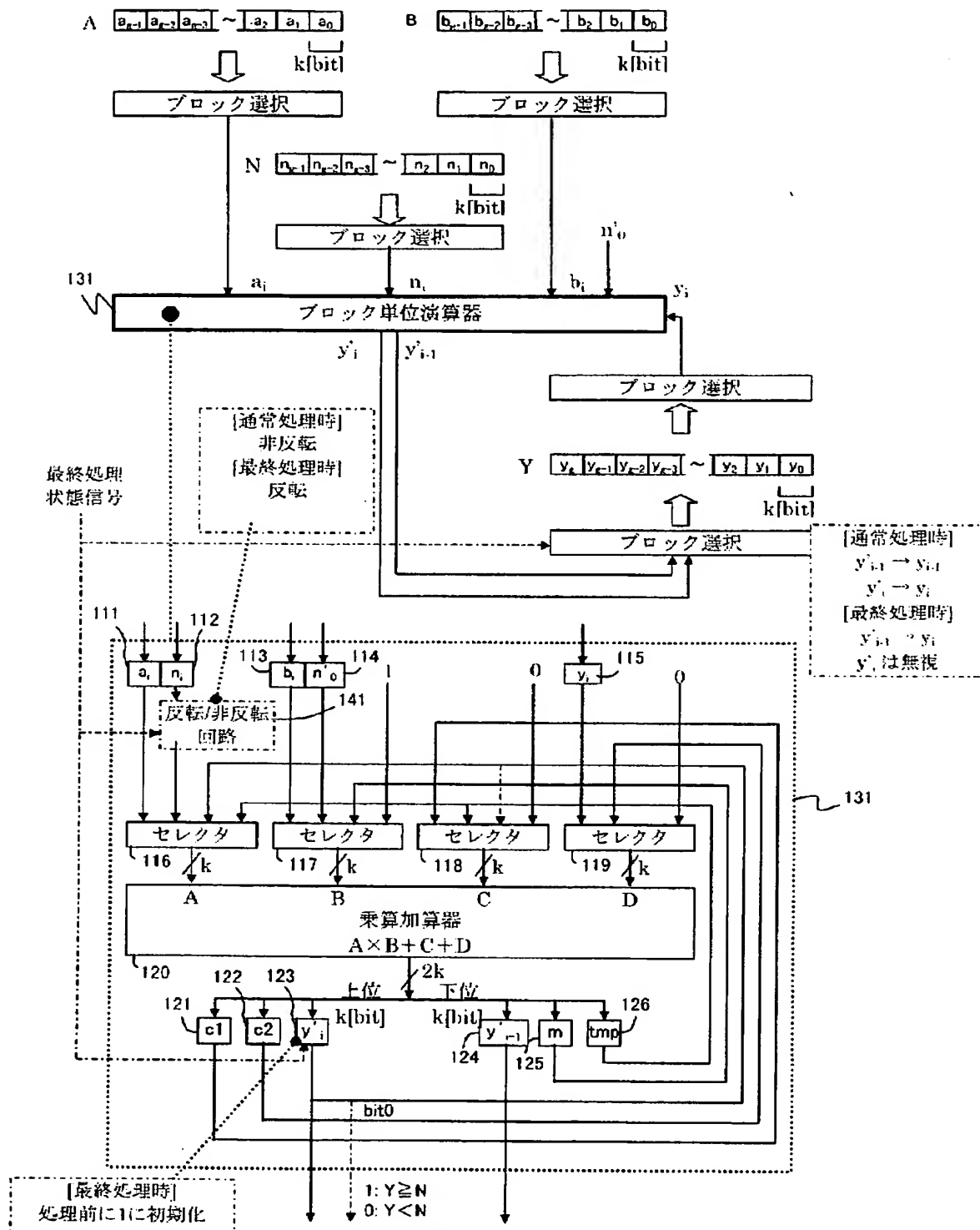
【図 14】

多倍長減算を示す図



【図 15】

仮想的な多倍長演算回路を示す図



【書類名】 要約書

【要約】

【課題】 モンゴメリ乗算剰余の多倍長演算を行う回路において、ブロック単位演算器における減算のための遅延時間を短縮し、動作周波数を維持したままで演算を行う。

【解決手段】 補数モードにおいて、2 次 B o o t h アルゴリズムのエンコーダ手段 2 0 2 は、乗算 $A \times B$ を行う通常モードとは異なる選択信号を出力し、選択手段 2 0 3 は、B の 3 つのビット b_1 、 b_0 、および b_{-1} に対しては $-A$ を表す部分積を選択し、それ以外のビットに対しては 0 を表す部分積を選択する。加算手段 2 0 4 は、 $-A$ を表す部分積と 0 とを加算して、 $-A$ (A の 2 の補数または A の 1 の補数) を出力する。

【選択図】 図 1

特願 2003-046527

出願人履歴情報

識別番号

[000005223]

1. 変更年月日

1996年 3月26日

[変更理由]

住所変更

住 所

神奈川県川崎市中原区上小田中4丁目1番1号

氏 名

富士通株式会社